МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет “Львівська політехніка”

Інститут ІКНІ

Кафедра ПЗ



**ЗВІТ**

До лабораторної роботи № 3

З дисципліни : “Вступ до інженерії програмного забезпечення ”

**Лектор:**

ст. викл. каф. ПЗ

Левус Є.В.

**Виконав:**

ст. ПЗ-15

Марущак А.С.

**Прийняв**:

ассист. каф. ПЗ

Cамбір А. А.

“ ” \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

Львів-2022

**Тема.** Зміст та роль тестування у життєвому циклі розробки програмного забезпечення.

**Мета.** Вивчити основні методи та принципи тестування на прикладі створення власної програми.

**Теоретичні відомості**

**14.** **Поясніть, що означає думати про тестування на етапі визначення та аналізу вимог**

Це означає, що ще при створені всіх вимог до програми треба завчасно думати про те, чи ці вимоги можна буде протестувати. Бо, наприклад вимогу «Програма має мати красивий інтерфейс» об'єктивно не можна верифікувати.

Саме тому, всі вимоги, що містяться в ТЗ(або іншому документі, що відображає вимоги до ПЗ) мають бути верифіковними.

**21.** **Як визначити повноту структурного тестування?**

Треба удостовіритись в тому, що тестові дані змушують працювати програму у всіх логічних відгалуженнях коду. Тобто, умовно кажучи, треба пройтися всіма стежками, що передбачені кодом програми.

**47. Що означає властивість живучість ПЗ? Як її можна перевірити?**

Живучість – здатність системи функціонувати при вводі недопустимих даних або ж у напружених умовах. Задля її перевірки треба включити у тестові дані якомога більше неправильних та крайніх випадків.

**Постановка завдання**

1. Провести функціональне тестування програми для роботи з динамічними списками (розробленої у ході виконання лабораторної роботи з дисципліни «Основи програмування»), використавши такі тести:

1. димовий тест,

2. тест нормальних умов,

3. тест екстремальних(граничних) умов,

4. тест виняткових умов.

1.1. Результати тестів 1, 2, 3 та 4 оформити у вигляді таблиці (за зразок взяти таблицю 2). У таблиці інформація заноситься в зрозумілій формі, достатньо повно, щоб не вимагалося додаткових (усних) пояснень. Для представлення тестових даних та фактичних результатів бажано використовувати посилання на зображення екранів чи відповідні файли. В очікуваних результатах зафіксувати, яка функція була протестована та відповідно, що є очікуваним результатом. Якщо помилки не знайдено у Тип звіту ставимо знак -. Примітки використовуємо для додаткових пояснень.

1.2. Знайдені дефекти виправити. Провести повторне тестування, результати якого оформити окремою таблицею.

2. Окремо провести тестування логічної схеми програми (головна програма та дві функції-підпрограми). Для цього зобразити її у вигляді графів управління. Вершини графів пронумерувати. Записати усі отримані маршрути роботи програми, використовуючи номери вершин.

2.1. Для кожного маршруту провести тестування і записати результати тестування у звітній таблиці.

2.2. Знайдені дефекти виправити. Провести повторне тестування, результати якого оформити окремою таблицею..

**Виконання роботи**

1. **Результати функціонального тестування:**

**Результати функціонального тестування програми**

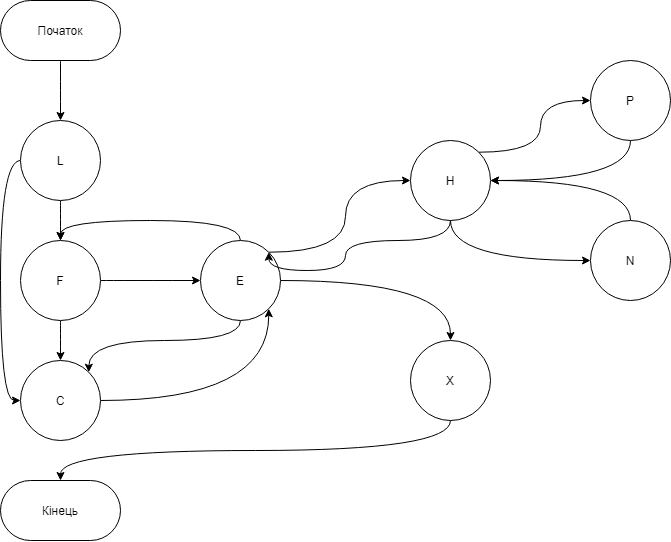
**LibraryUtil, версія 0.0.1**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ з/п** | **Тестові дані** | **Фактичні результати** | **Очікувані результати** | **Ступінь критичності** | **Тип звіту** | **Примітки** |
| **Димове тестування** | | | | | | |
| **1** | - | Запуск програми без помилок | Запуск програми без помилок | - | - | - |
| **2** | - | Завершення роботи без помилок | Завершення роботи без помилок | - | - | - |
| **Тест нормальних умов** | | | | | | |
| **1** |  | Нормальна робота команд. Але при наявності крапки з комою в кінці стрічки вважає його за порожню команду | Правильна обробка скриптових файлів з різними функціями | Не критично | Verified | Помилку викликає наявність пробілу після крапки з комою. |
| **2** |  | Правильне зчитування правильних вхідних даних з файлу | Правильне зчитування правильних вхідних даних з файлу | **-** | **-** | **-** |
| **3** |  | Правильне сортування правильних даних за всіма ознаками | Правильне сортування правильних даних за всіма ознаками | **-** | **-** | **-** |
| **4** |  | Правильна фільтрація правильних даних за всіма ознаками | Правильна фільтрація правильних даних за всіма ознаками | **-** | **-** | **-** |
| **5** |  | Правильне обмеження списку до певної к-ті книг | Правильне обмеження списку до певної к-ті книг | **-** | **-** | **-** |
| **6** |  | Створення декількох списків паралельно | Створення декількох списків паралельно | **-** | **-** | **-** |
| **7** |  | Перемикання між списками книг за звичайних умов | Перемикання між списками книг за звичайних умов | **-** | **-** | **-** |
| **8** |  | Правильне додавання книг вручну | Правильне додавання книг вручну | **-** | **-** | **-** |
| **9** |  | Вивід списку достуних списків книг | Вивід списку достуних списків книг | **-** | **-** | **-** |
| **10** |  | Вивід к-ті книг у поточному списку | Вивід к-ті книг у поточному списку | **-** | **-** | **-** |
| **11** |  | Збереження списку у текстовий файл | Збереження списку у текстовий файл | **-** | **-** | **-** |
| **12** |  | Помилка при видалені одного порожнього списку | Нормальне видалення списків | Критична | Verified | Якщо списки не порожні, то проблеми не виникає |
| **13** |  | Отримання довідки по командам оболонки | Отримання довідки по командам оболонки | **-** | **-** | **-** |
| **Тест екстремальних(граничних) умов** | | | | | | |
| **1** |  | Нормальне опрацювання порожніх текстових файлів(або з порожніми стрічками) з виводом відповідного повідомлення | Нормальне опрацювання порожніх текстових файлів(або з порожніми стрічками) з виводом відповідного повідомлення |  |  |  |
| **2** |  | Нормальне опрацювання великих за обсягом тектових файлів | Нормальне опрацювання великих за обсягом тектових файлів |  |  |  |
| **3** |  | Помилки не виникає, проте і повідомлення про сорутвання порожнього списку немає | Сортування порожнього списку з виводом відповідного повідомлення | Не критично | Verified | - |
| **4** |  | Помилки не виникає, проте і повідомлення про фільтрацію порожнього списку немає | Фільтрація порожнього списку з виводом відповідного повідомлення | Не критично | Verified | - |
| **5** |  | Обмеження порожнього списку з виводом відповідного повідомлення | Обмеження порожнього списку з виводом відповідного повідомлення | **-** | **-** | **-** |
| **Тест критичних умов** | | | | | | |
| **1** |  | Нормальна обробка неправильної команди в скріпті з виводом відповідного повідомлення | Нормальна обробка неправильної команди в скріпті з виводом відповідного повідомлення |  |  |  |
| **2** |  | Нормальна обробка неправильного параметру команди з виводом відповідного повідомлення | Нормальна обробка неправильного параметру команди з виводом відповідного повідомлення |  |  |  |
| 3 |  | Помилка при передачі 2ох параметрів в функцію сортування | Нормальна обробка неправильної к-ті параметрів команди скрпіта з виводом відповідного повідомлення | Критично | Verified | - |
| 4 |  | Нормальна обробка запуску неіснуючого скриптового файлу з виводом відповідного повідомлення | Нормальна обробка запуску неіснуючого скриптового файлу з виводом відповідного повідомлення |  |  |  |
| 5 |  | Нормальна обробка зчитування з неіснуючого текстового файлу з виводом відповідного повідомлення | Нормальна обробка зчитування з неіснуючого текстового файлу з виводом відповідного повідомлення |  |  |  |

Результати повторного тестування

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ з/п** | **Тестові дані** | **Фактичні результати** | **Очікувані результати** | **Ступінь критичності** | **Тип звіту** | **Примітки** |
| 1 |  | Правильна обробка скриптових файлів з різними функціями | Правильна обробка скриптових файлів з різними функціями | - | - | - |
| 2 |  | Нормальне видалення списків | Нормальне видалення списків | - | - | - |
| 3 |  | Сортування порожнього списку з виводом відповідного повідомлення | Сортування порожнього списку з виводом відповідного повідомлення | - | - | - |
| 4 |  | Фільтрація порожнього списку з виводом відповідного повідомлення | Фільтрація порожнього списку з виводом відповідного повідомлення | - | - | - |
| 5 |  | Нормальна обробка неправильної к-ті параметрів команди скрпіта з виводом відповідного повідомлення | Нормальна обробка неправильної к-ті параметрів команди скрпіта з виводом відповідного повідомлення | - | - | - |

**Логічна схема програми у вигляді графів управління**



L – очистка екрану

F – виконання скрпітового файлу

C – виконання в консолі

E – виконання однієї команди

H – отримання довідки по команді

P – існуюча команда

N – неіснуюча команда

X – вихід з програми

Результати структурного тестування

|  |  |  |
| --- | --- | --- |
| № з/п | Маршрут | Результат |
| 1 | (Початок)-L-F-E-X-(Кінець) | Успішне виконання |
| 2 | (Початок)-L-F-E-H-P-E-X-(Кінець) | Успішне виконання |
| 3 | (Початок)-L-F-E-H-N-E-X-(Кінець) | Успішне виконання |
| 4 | (Початок)-L-C-E-X-(Кінець) | Успішне виконання |
| 5 | (Початок)-L-C-E-H-P-E-X-(Кінець) | Успішне виконання |
| 6 | (Початок)-L-C-E-H-N-E-X-(Кінець) | Успішне виконання |
| 7 | (Початок)-L-F-E-F-C-E-X-(Кінець) | Успішне виконання |
| 8 | (Початок)-L-F-E-F-C-E-H-P-E-X-(Кінець) | Успішне виконання |
| 9 | (Початок)-L-F-E-F-C-E-H-N-E-X-(Кінець) | Успішне виконання |
| 10 | (Початок)-L-F-E-H-P-E-F-C-E-X-(Кінець) | Успішне виконання |
| 11 | (Початок)-L-F-E-H-P-E -F-C-E-H-P-E-X-(Кінець) | Успішне виконання |
| 12 | (Початок)-L-F-E-H-P-E -F-C-E-H-N-E-X-(Кінець) | Успішне виконання |
| 13 | (Початок)-L-F-E-H-N-E-F-C-E-X-(Кінець) | Успішне виконання |
| 14 | (Початок)-L-F-E-H-N-E -F-C-E-H-P-E-X-(Кінець) | Успішне виконання |
| 15 | (Початок)-L-F-E-H-N-E -F-C-E-H-N-E-X-(Кінець) | Успішне виконання |

**Висновок:** Виконавши цю лабораторну роботу, ми вивчили основні методи та принципи тестування на прикладі створенної власної програми. Мною особисто було знайдено декілька критичних і не критичних помилок, тому тестування можна вважати успішним.